Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-11 (Canceled)

12. (Currently Amended) A method ~~of fabricating an integrated circuit implementing multiple program operations~~, comprising:

deriving control flow graphs for selected multiple program operations of a source code;

identifying basic blocks of the control flow graphs;

developing data flow graphs for two or more of the basic blocks;

identifying a common subgraph shared by at least a pair of the basic blocks including identifying seed basic blocks by identifying candidate seed basic blocks among the basic blocks of the control flow graphs, and comparing candidate seed basic blocks from control flow graphs of separate program operations;

scheduling the shared processes represented by the common subgraph;

scheduling the shared processes for operation in each of the multiple program operations; and

scheduling of processing units to carry out the common subgraph.~~, including:~~

~~clustering the shared processes into a macroblock having nodes representing the shared processes and at least a plurality of unconditional, conditional, and reconfiguration edges running between nodes;~~

~~determining a relative delay among possible paths through the common subgraph;~~

~~performing branch and bound scheduling for the longest-delay-time path, and for less than all possible paths through the common subgraph including at least a longest-delay-time path; and~~

~~merging all schedules; and~~

~~laying out an arrangement of circuit elements for implementation of the integrated circuit in hardware, including:~~

~~grouping the circuit elements into first level clusters; and~~

~~placing the first level clusters by grouping the first level clusters together to form second level clusters and placing the second level clusters.~~

13. (Canceled)

14. (Currently Amended) The method of claim ~~13~~12, wherein said identifying seed basic blocks includes identifying ones of the basic blocks that lie inside a loop.

15. (Previously Presented) The method of claim 14, wherein said identifying ones of the basic blocks that lie inside a loop includes identifying one of:

a single nested level loop with only one basic block;

a single nested level loop with more than one basic block; and

a multi-level nested loop.

16. (Previously Presented) The method of claim 14, wherein said identifying ones of the basic blocks that lie inside a loop includes identifying one of:

a single nested level loop with more than one basic block; and

a multi-level nested loop.

17. (Previously Presented) The method of claim 16, wherein said identifying seed basic blocks further includes identifying basic blocks of control flow graphs of separate program

operations under like control.

18. (Previously Presented) The method of claim 17, wherein said identifying seed basic blocks further includes determining a count of each operation type in a basic block.

19. (Previously Presented) The method of claim 18, wherein said identifying seed basic blocks further includes examining edges in a data flow graph of candidate seed basic blocks of control flow graphs from the separate programming operations.

20. (Previously Presented) The method of claim 19, wherein said examining edges includes classifying edges based on source and destination node operation type.

21. (Previously Presented) The method of claim 20, wherein said examining edges includes eliminating edges of one data flow graph having a source-operation-to-destination-operation not found in another data flow graph having edges under examination.

22. (Previously Presented) The method of claim 21, further comprising implementing the eliminated edges in a circuit other than in an application specific integrated circuit.

23. (Previously Presented) The method of claim 22, wherein said implementing the eliminated edges includes implementing the eliminated edges in one or more look up tables.

24. (Previously Presented) The method of claim 20, wherein said examining edges further includes comparing associativity among edges being compared.

25. (Previously Presented) The method of claim 24, wherein said comparing associativity includes determining numbers of predecessor, siblings, companions, and successors of edges being compared.

26. (Previously Presented) The method of claim 12, wherein said scheduling the shared processes represented by the common subgraph includes ASAP scheduling the common

subgraph.

27. (Previously Presented) The method of claim 12, further comprising providing common operations of the common subgraph in an application specific integrated circuit.

28. (Previously Presented) The method of claim 12, further comprising:

identifying at least one other common subgraph shared by the at least a pair of the basic blocks;

scheduling other shared processes represented by the other common subgraph;

scheduling the other shared processes for operation in each of the multiple program operations; and

laying out another arrangement of other circuit elements for implementation of the integrated circuit in hardware, including:

grouping the other circuit elements into other first level clusters; and

placing the other first level clusters by grouping the other first level clusters together to form other second level clusters and placing the other second level clusters.

29. (Previously Presented) The method of claim 12, wherein said identifying a common subgraph shared by at least a pair of the basic blocks includes identifying a largest common subgraph shared by the at least a pair of the basic blocks.

30. (Previously Presented) The method of claim 12, wherein said scheduling the shared processes represented by the common subgraph includes providing switching of differing delays among processes of the common subgraph to effect subgraphs operating each of the selected multiple program operations.

31. (Previously Presented) The method of claim 30, wherein said providing switching includes providing multiplexers operative to apply alternative delays between processes of

the common subgraph.

32. (Previously Presented) An integrated circuit fabricated by the method of claim 12.

33. (Previously Presented) A computer-readable medium comprising stored programming instructions which, in response to execution by a processor of an apparatus, causes the apparatus to perform the method of claim 12.

34 – 62 (Canceled)

63. (New) The method of claim 12, wherein said scheduling of processing units to carry out the common subgraph includes:

clustering the shared processes into a macroblock having nodes representing the shared processes and at least a plurality of unconditional, conditional, and reconfiguration edges running between nodes;

determining a relative delay among possible paths through the common subgraph;

performing branch and bound scheduling for the longest-delay-time path;

merging all schedules; and

laying out an arrangement of circuit elements for implementation of the integrated circuit in hardware, including:

grouping the circuit elements into first level clusters; and

placing the first level clusters by grouping the first level clusters together to form second level clusters and placing the second level clusters.